

Memoria del proyecto de manipulación de imágenes .pgm mediante C++

Alejandro Villanueva Prados

1. Descripción General del Proyecto

Este proyecto PGM, el cual lo comenzamos con un struct y diversas funciones, conforme avanzamos en conocimientos en la asignatura, lo mejoramos implementando una clase con diversas funciones miembro.

La clase PGM tiene como datos miembro de la clase tenemos 3 short int que hacen referencia a el alto, el ancho y a cada elemento de una matriz 650 x 650.

Hemos desarrollado las siguientes funciones miembro, las cuales comentaremos aquí brevemente:

- Función miembro de entrada de datos. Se leen el identificador de archivo PGM (que es P2), el alto, el ancho, el valor maximo que puedo tomar el blanco (el 255).
- Función miembro de salida de datos. Va escribiendo sucesivamente el identificador, el valor del ancho, del alto, y el el valor de cada componente de la matriz que va en el rango de 0 a 255.
- Función miembro Blanquear. Pasa todos los valores de la matriz a 255, que es el valor máximo de luminosidad (el “blanco”).
- Función miembro Contraste Máximo. Si un valor de pixel de imagen esta por debajo de la mitad (127) se manda a 0 (oscuridad máxima), en caso de que sea mayor o igual a la mitad, se manda a 255, que es mandar

2. Descripción del trabajo de cada integrante

2.1. Francisco Bonillo González

Mi parte del trabajo se basó principalmente en implementar una función miembro capaz de calcular el “negativo” de una imagen dada como entrada

del programa.

Esto es, mediante dos bucles anidados según el alto y el ancho de una imagen, se va recorriendo cada componente de la matriz, que conforma cada uno de los píxeles de la imagen, y se va cambiando a su vez el número asignado que tenía (entre 0 y 255), por su complementario.

Por ejemplo, en el caso de que el píxel tuviera un valor de x , se calcularía pues el negativo – su complementario – que es $255 - x$. Ese x es el componente `pixel[i][j]` de la matriz de la imagen. Así se varía y se invierte la luminosidad de manera que la imagen queda en negativo.

Además, mi implicación en el trabajo ha sido de ayudante tanto en la sección de elaboración de memoria como en el desarrollo de otras funciones más generales.

2.2. Miguel Piñar Pérez

Mi trabajo consistió en:

- El struct inicial que almacenaba el ancho, el alto, y una matriz de 650x650 que almacenaba los valores de los píxeles de la imagen.
- La función de blanqueo. Esta función recorre toda la matriz cambiando el valor de todos los píxeles de la matriz a 255 (blanco).
- La función para trasponer una matriz, primera parte de la función de rotación. Para rotar una matriz 90° a la derecha hay que trasponerla y después cambiar sucesivamente sus columnas: la primera por la última, la segunda por la penúltima... La función de trasposición recorre una mitad de la matriz intercambiando los valores (i,j) por los (j,i) .

2.3. Alejandro Villanueva Prados

En este proyecto, mi trabajo ha consistido en gestionar el control de versiones, haciendo uso de la plataforma GitHub. Así todos los miembros hemos dispuesto de un lugar común para subir el código en el que trabajamos. Aparte también he redactado esta memoria, organizando las memorias del resto de los integrantes, añadiendo mi parte, y dar un formato a todo usando \LaTeX

Con respecto a mi aportación al código, he implementado la función de salida de datos y también convertí el código de una versión anterior, basada en un struct, a una clase con diversos métodos, ya que así incluimos los últimos conocimientos adquiridos en la asignatura y también simplifica el

código. Luego realicé la función ‘Swap’ que intercambia dos columnas dadas de la matriz. Por último me encargué de documentar el proyecto usando Doxygen.